Abstract Title: Towards Algebraic Query Optimization in Process Data Warehouse

Ying-Feng Hsu Vladimir Zadorozhny
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260
E-mail: {yfhsu, vladimir} @sis.pitt.edu

ABSTRACT
We consider a novel query processing approach for data analysis in Process Data Warehouse system that performs summarization and discovery of trends in dynamic data from complex processes. In particular, we introduce an algebraic framework that can be used as a basis for algebraic optimization in PDW system.

General Terms
Algorithms, Experimentation, Theory

Keywords
Pattern, Markov Chain

1. INTRODUCTION
The amount and value of data available due to rapid spread of information technology is exploding. Typical large enterprises have approximately a petabyte of operational data stored in hundreds of data repositories supporting thousands applications. Data storage volumes grow in excess of 50% annually. In order to cope with the predicted growth rates the task of quick summarization and utilization of data becomes urgent.

In this research we consider a novel Process Data Warehouse (PDW) technology that implements a holistic declarative paradigm for summarization and discovery of trends in dynamic data from complex processes. The proposed approach includes a method, system architecture and optimization techniques for efficient summarization of large data sets. A unique feature of the proposed approach is that the summarization schema is independent of the source data dimensionality, which makes the proposed technology highly scalable and applicable for a wide class of data management tasks. In addition, PDW uses adaptive techniques tuning its performance to accommodate high data load.

Potential uses of the proposed approach range from efficient data utilization in specialized data centers to Internet-scale data search and analysis engines. For instance, a PDW system can explore the most and least probable dynamic scenarios in complex processes that generate large amounts of raw data. Consider a drug study process associated with large-scale data collecting activities. An example of the PDW analysis would be estimating complex scenarios in patient development, such as "it is most likely with probability of 0.8 that a two-week period of improvement in a patient condition will be followed by a three day period of increase in blood pressure, after which the patient will start feeling better up to complete recovery in ten days. However, there is also a less probable scenario (probability of 0.2) with no indication of improvement for three weeks followed by considerable aggravation in the patient's condition." Similar scenario can be utilized in many other application domains (e.g., genome data analysis, market data monitoring, natural disasters and structural health data analysis, etc.). At the same time PDW can scale to very high data loads, processing large amounts of raw data and performing multiple concurrent requests.

IConference'09, Feb 8–11, 2009, Chapel Hill, NC, USA.

2. PDW ARCHITECTURE
A general PDW architecture is shown in Figure 1. The PDW system performs a multistage summarization of large data streams in order to analyze dynamic trends (pattern) in the environment that has produced the input data stream. The summarization schema is based on a process model that can be represented as a state machine $S = (Q;\Sigma;\delta)$, where Q is a set of states, Σ is a set of observable events or observations, and δ is a set of possible associations between the states and observable events. The only user input is a set of mapping rules that defines application-dependent states and observations of interest.

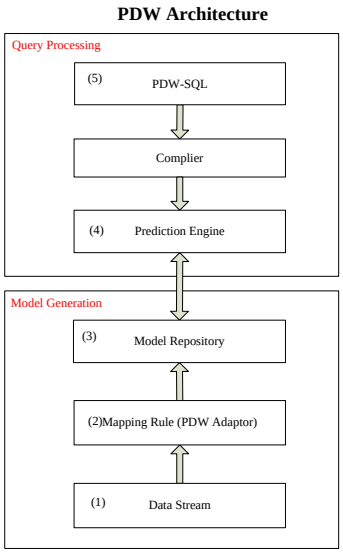

Figure 1: General PDW Architecture

Mapping rules assign symbolic state and observation names to logical constraints over the input data streams. Using the mapping rules, the PDW system automatically generates data transformation procedures that constitute a PDW Data Stream Adaptor. The PDW Adaptor produces an output stream of states and observation. Finally, PDW uses dynamic process model to generate probabilistic assessments of the events in response to Decision Support Queries specified in PSQL, an SQL-like declarative stream query language.

To sum up, raw operational process data is continually collected by lower layer of the PDW system. The PDW system utilizes that data to create process model and to perform data summarization and analysis using decision-support queries. In this poster, we will focus on the PDW algebra that forms a core for query optimization in the PDW query processing (prediction) engine.

3. PDW Algebra

Under the classical DBMS query processing architecture a query in a high-level declarative language, such as SQL, is converted to an efficient query execution plan. The query plan is an operational tree with assigned physical implementations of algebraic operators. In the process of cost-based optimization, the optimizer estimates a cost of query plan and aims to generate the least expensive plan.

The efficient algebraic cost-based optimizer is a core component of DBMS. We aim at accommodating algebraic query optimization to improve performance of PDW decision support queries. To do this we define PDW algebra that optimizer will utilize to generate efficient query execution plans in PDW system. The signature of PDW algebra includes a set of types together with operations defined on them. Our PDW algebra includes data types specifically designed to support PDW process data model. Examples of such types are sequence type and pattern, which we briefly consider below.

- **Sequence type.** Sequence type represents an ordered collection of states (e.g. a temperature state sequence <High, Low, Low, Moderate>). This type allow us to explore probabilities of the state sequences in chronological order. In this research, we utilize first-order Markov assumption, that is the probability of a certain state (Sn) at time n depends only on the state (Sn-1) at time n - 1.

- **Pattern type.** Unlike Sequence type that focuses on representing sequence of events, pattern type represents more general trends within the observed domain. For example, we may look for probability of a non-decreasing temperature over three days with no interest in the actual temperature values. Both <Low, Low, Moderate, High> and <Moderate, Moderate, High, High> sequences satisfy this non-decreasing pattern. In contrast, <High, High, Moderate, Moderate> does not satisfy that pattern since this sequence includes a temperature decrease from 'High' to 'Moderate'. Another reason to use Pattern type is to facilitate exploration of long observation sequences. The probability of individual sequences would naturally decrease as the sequence length becomes longer. This phenomenon might lead to ignoring important long-term trends. Pattern data type characterizes all state transition sequences that satisfy a common trend and thus explore a cumulative effect of multiple weaker trends. Since sequence satisfies a pattern and a pattern subsumes a sequence, the total probability of a pattern is calculated as a composition of probabilities of its corresponding sequences.

4. EXPERIMENTS

In this poster, we present our experiment results in two subsections. First, we focus on individual sequence analysis and compare several PDW analysis techniques. Examples of the considered analysis include finding the most probable event sequence and assessing probability of a generic patter.

First, we discuss event sequence analysis methods:

(1) BF (brute Force) method that calculates probabilities of all sequences to select the most probable sequence.

(2) MAXPS operation that implements a version of Viterbi algorithm and recursively explores the sequence space.

Then we explore performance of several pattern analysis methods:

(1) BF (brute Force) method evaluates probabilities of all sequences satisfying the pattern and combine them to estimate pattern probability.

(2) PS (Partial Sequence) approach approximates the pattern probability by using sequences satisfied by less general pattern subsumed by the original pattern.

(3) RR (Recursive Reuse) estimates the pattern probability without explicit enumeration of sequences that it subsumes. It considerably over-performs both BF and PS methods.

5. CONCLUSION

We proposed a novel query processing approach for data analysis in Process Data Warehouse system. We introduced and tested an algebraic framework that can be used as a basis for algebraic optimization in PDW system. One of the major contributions of this research is developing efficient approaches to the general pattern analysis in large-scale dynamic data stream.

Currently we are implementing an advanced prototype of PDW that we plan to apply to data analysis in a number of real word problems. We work on a high level PDW query language and a complier that would map the user queries in efficient algebraic query plans using the proposed PDW algebra. We also work on lower layers of the PDW architecture. In particular, we develop methods and tools to optimize generation of mapping rules that aggregate raw data stream in a number of states and observations of interest. We explore information theoretical concepts to evaluate quality of generated models in order to interpret raw streaming data in the most appropriate way.